

REMARKS

Applicants have studied the Office Action dated February 9, 2007, and have made amendments to the claims. It is submitted that the application, as amended, is in condition for allowance. By virtue of this amendment, claims 1-21 are pending. Claims 1, 6, 9, 14, and 17 have been amended, and new claim 21 has been added. Reconsideration and allowance of the pending claims in view of the above amendments and the following remarks are respectfully requested.

Claims 1, 11, and 17 were objected to for being unclear. Applicants have amended claims 1, 9, and 17 to remove the language that was objected to by the Examiner. This language is not present in claim 11. Therefore, it is respectfully submitted that the objection to claims 1, 11, and 17 should be withdrawn.

Claims 17-20 were rejected under 35 U.S.C. § 112, first paragraph, as not being enabled by the disclosure. Claim 17 has been amended as suggested by the Examiner. Therefore, it is respectfully submitted that the rejection of claims 17-20 under 35 U.S.C. § 112, first paragraph, should be withdrawn.

Claims 1, 2, 5-10, and 13-18 were rejected under 35 U.S.C. § 102(e) as being anticipated by Skinner et al. (U.S. Patent Application Publication No. 2003/0204517). Claims 3, 4, 11, 12, 19, and 20 were rejected under U.S.C. § 103(a) as being unpatentable over Skinner et al. in view of Hutsch et al. (U.S. Patent Application Publication No. 2001/0034771). These rejections are respectfully traversed.

The present invention is directed to an efficient and easy-to-implement method for managing configuration data. One embodiment of the present invention provides a method for managing configuration data. According to the method, configuration values are stored in a hierarchical tree having multiple nodes, a defined structure, and defined data types for the stored configuration values. Each node is associated with at least one of the configuration values, and each of the configuration

values dictates how an application component associated with that configuration value behaves and/or interacts with other application components. Additionally, some of the nodes are only associated with a set of configuration values, while other of the nodes are associated with a combination of a set of configuration values and an identifier associated with at least one application component. An application component is registered with at least one of the nodes of the tree, based on a query received from the application component. The application component is notified when a configuration value associated with the at least one node of the tree is modified, based on an addition or change in at least one configuration value that matches the query.

For example, in the exemplary embodiment disclosed in the specification, configuration data, which includes configuration values, is organized into a hierarchical tree. Application components register interest in a particular node of the tree, or a particular sub-tree. Because they registered, these components receive notifications whenever a corresponding configuration value changes. Thus, any component that is a current holder of a registration or reference will be notified when a configuration value changes. Because the configuration values are arranged hierarchically in a tree such as an XML tree, the various application components can register for callbacks or notification upon modification of any nodes in any sub-tree.

Further, the structure of the configuration data in the XML tree can be altered, for example by adding more sub-nodes to a particular node in the tree. An interested party to the changed sub-tree will receive the new configuration data. Thus, an application component can register itself as an interested party to any change in a configuration value. Also, the use of XML to represent the tree allows a sub-tree of configuration data to be easily be expanded, with no change in how an application component handles those configuration values. Thus, an application component (such as a Graphical User Interface, an object, an Application Programming Interface, a plug-in, an application, or a user) has the ability to handle changing configuration data, and can handle configuration data in a hierarchical structure such as by using the extensibility of XML.

The Skinner reference is directed toward object oriented programming and performing active update notifications within a multi-tiered application. The active update notification process of Skinner requires application components to register "interest objects". "Interest objects" indicate an interest of an application component with respect to a data object or set of data objects. The interest object is registered with an update management component of the application. The interest object specifies the interested application component, as well as the identity of one or more data objects or an attribute value or range of values to associate with data objects. When modifications are made to data objects corresponding to the registered interest objects, the interested application component (or components) receives an update notification from the update management component.

The interests of an application component can be further refined by registering other interest objects as sub-interests of existing interest objects. This results in an interest hierarchy within an interest registry. Skinner also teaches that interested components, also referred to as "observers," can be registered for each interest object in the hierarchy to receive notification of updates. One type of interest object collects data objects that are part of a changed data set. This allows an interested component to access those collected data objects after receipt of an update notification.

Amended claim 1 recites:

storing a plurality of configuration values in a hierarchical tree having a plurality of nodes, a defined structure, and defined data types for the stored configuration values, wherein each node is associated with at least one of the configuration values, and each of the configuration values dictates how an application component associated with that configuration value at least one of behaves and interacts with other application components, and wherein some of the nodes are only associated with a set of configuration values while other of the nodes are associated with a combination of a set of configuration values and an identifier associated with at least one application component;

registering at least one application component with at least one of the nodes of the tree, based on at least one query received from the at least one application component; and

notifying the at least one application component when a configuration value associated with the at least one node is modified, based on an addition or change in at least one configuration value that matches the at least one query.

Amended claims 9 and 17 contain similar recitations.

In contrast to what is recited in the amended claims, Skinner explicitly teaches two interest registries: a server interest registry and a client interest registry. These registries are trees comprising interest objects, not configuration values as recited in the amended claims. The Examiner states that Skinner teaches “storing a plurality of configuration values [data objects] in a hierarchical tree having a plurality of nodes (Fig 5B), a defined structure [schema] (see [0122]-[0123], and defined data types (see [0045]-[0047]) for the stored configuration values, wherein each node is associated with at least one of the configuration values (see [0061])”.

However, at the portions of Skinner cited by the Examiner (and throughout Skinner), two registries are taught: a server registry residing at an application server and a client registry residing at a client device. The server interest registry at the application server has various types of nodes, with one type of node only identifying a client. The server interest registry residing on the application server is different than the claimed limitation of “storing a plurality of configuration values in a hierarchical tree having a plurality of nodes . . . , wherein each node is associated with at least one of the configuration values.” FIG. 5B of Skinner, which is referred to by the Examiner, illustrates the client interest registry that resides at the client.

The server interest registry at the application server and the client interest registry at the client are only concerned with interest objects. The “interest object” used in Skinner “specifies the interested application component, as well as the identity of one or more data objects or an attribute value or range of values to associate with data objects”. See Skinner at Abstract. Therefore, assuming arguendo that Skinner’s interest registry is a hierarchical tree, it is a hierarchal tree of interest objects, not configuration values as recited in the amended claims.

The “interest objects” of Skinner comprise interest criteria that are used to determine if an update notification is of any interest. For example, when a data object is changed, the root node 504 of the client interest registry is notified. The root node 504 passes the update notification to each of its registered interest objects 506, 507, 508, 510, 511, and 513. Each interest object then checks its

interest criteria to determine whether or not the update notification is for a data object that they are interested in. See Skinner at paragraphs 0077-0084. Therefore, Skinner teaches that only data objects interested in by an application component have an associated interest object in the interest registries.

In embodiments of the present invention, on the other hand, “some of the nodes are only associated with a set of configuration values while other of the nodes are associated with a combination of a set of configuration values and an identifier associated with at least one application component”. In other words, configuration values can be stored within the hierarchical tree without being associated with an interested application component. Nowhere does Skinner teach this. The entire focus of Skinner’s interest registries is to only include interest objects that by definition indicate an application component’s interest in a data object. Therefore, the amended claims distinguish over Skinner.

Further, the Examiner states that Skinner’s “data objects” are equivalent to the recited “configuration values”. This would require Skinner to teach, among other things, a hierarchical tree that comprises a plurality of nodes, with each node associated with at least one data object, and a data object can exist in the hierarchical tree without being associated with an interested application component. However, Skinner does not teach this. Therefore, the amended claims distinguish over Skinner.

The Examiner also states that Skinner teaches the recited limitation that each of the configuration values dictates how an application component associated with that configuration value at least one of behaves and interacts with other application components. According to the Examiner, Skinner teaches “the data objects contain attributes (variables), metadata attributes and values; since the values are located in a hierarchical tree, if a component registers to receive notification of a change to that variable, this is considered to represent the interaction] (see [0061]).” Applicants respectfully disagree. Paragraph 0061 of Skinner merely states:

In accordance with an embodiment of the invention, interest can be expressed for a single data object or for a set of data objects. Interest for a single data object may be expressed explicitly, such as by specifying a unique identifier of the data object such as an object ID or serial number. Interest for a set of data objects may be expressed either explicitly with a set of unique object identifiers, or the interest may be expressed in terms of interest criteria. Each data object comprises one or more data attributes (or variables), and may further contain one or more metadata attributes. Metadata attributes comprise data or descriptions about a data object or the attributes the data object contains. Interest criteria may include a specific value or range of values for one or more attributes of a data object. To evaluate whether a data object meets the interest criteria, the criteria is tested against attributes of the data object.

Nowhere does Skinner teach that “each of the configuration values dictates how an application component associated with that configuration value at least one of behaves interacts with other application components”. Skinner only states that “the data objects contain attributes (variables), metadata attributes and values”. This conflicts with the Examiner’s assertion that “if a component registers to receive notification of a change to that variable, this is considered to represent the interaction”.

Applicants believe that the differences between Skinner and the present invention are clear in amended claims 1, 9, and 17, which set forth various embodiments of the present invention. Therefore, claims 1, 9, and 17 distinguish over the Skinner reference, and the rejection of these claims under 35 U.S.C. § 102(e) should be withdrawn.

As discussed above, amended claims 1, 9, and 17 distinguish over the Skinner reference. Furthermore, the claimed features of the present invention are not realized even if the teachings of Hutsch are incorporated into Skinner. Hutsch does not teach or suggest the claimed features of the present invention that are absent from Skinner. Thus, claims 1, 9, and 17 distinguish over the Skinner and Hutsch references, and thus, claims 2-8, claims 10-16, and claims 18-20 (which depend from claims 1, 9, and 17, respectively) also distinguish over Skinner and Hutsch.

Furthermore, it is submitted that limitations in the dependent claims are absent from Skinner and Hutsch. For example, with respect to amended claims 6 and 14, the Examiner states that Skinner teaches “modifying at least one configuration value (see [0058], line 3); storing in the hierarchical

tree the configuration value that was modified (see [0066], lines 1-5); and notifying the at least one application component that the configuration value was modified (see [0058], lines 7-9)." However, amended claim 6 recites:

modifying at least one configuration value that is associated with the at least one node with which the at least one application component is registered;

storing in the hierarchical tree the configuration value that was modified, the configuration value being stored at the at least one node with which the at least one application component is registered; and

notifying the at least one application component that the configuration value was modified.

Amended claim 14 contains similar recitations.

Paragraph 0058, line 3 of Skinner merely states that the interest registry is notified when a change to a data object is made. Paragraph 0066, lines 1-5 merely state that "LiveCollectedUpdates 402 is configured to collect data objects associated with changes made during a transaction, effectively caching those changes". The phrase "caching those changes" is very different from "storing in the hierarchical tree the configuration value that was modified". Nowhere does Skinner teach that the modified value is stored in the hierarchical tree. Applicants have amended claims 6 and 14 for clarity. In view of these amendments, the differences with Skinner should be very clear.

Therefore, it is respectfully submitted that the rejections of claims 1-20 under 35 U.S.C. § 102(e) and 35 U.S.C. § 103(a) should be withdrawn.

No amendment made was related to the statutory requirements of patentability unless expressly stated herein. No amendment made was for the purpose of narrowing the scope of any claim, unless Applicants have argued herein that such amendment was made to distinguish over a particular reference or combination of references.

Claim 21 has been added by this amendment, and is provided to further define the invention disclosed in the specification. Claim 21 is allowable for at least the reasons set forth above with respect to claims 1-20.